

REMARKS

Applicants have studied the Office Action dated November 29, 2007. Claims 21, 23, and 25 have been amended. No new matter has been added. It is submitted that the application is in condition for allowance. By virtue of this amendment, claims 3-13, 16-19, and 21-25 are pending. Reconsideration and further examination of the pending claims in view of the above amendments and the following remarks is respectfully requested. In the Office Action, the Examiner:

- Rejected claims 3, 7-11, 13, 16, 21, 23, 25 under 35 U.S.C. §103(a) as being unpatentable over Meyerzon et al. (U.S. Patent No. 6,638,314), in further view of Blumenthal (U.S. Patent No. 6,026,409), and in further view of Koike et al. (U.S. Patent No. 7,194,678);
- Rejected claims 4-6 and 17-19 under 35 U.S.C. 103(a) as being unpatentable over Meyerzon et al. (U.S. Patent No. 6,638,314), Blumenthal (U.S. Patent No. 6,026,409), and Koike et al. (U.S. Patent No. 7,194,678), and in further view of Hobbs (U.S. Patent No. 6,523,022); and
- Rejected claims 12, 22, and 24 under 35 U.S.C. 103(a) as being unpatentable over Meyerzon et al. (U.S. Patent No. 6,638,314), Blumenthal (U.S. Patent No. 6,026,409), and Koike et al. (U.S. Patent No. 7,194,678), and in further view of Lawrence (U.S. Patent No. 6,289,342).

Rejection under 35 U.S.C. §103(a) in view of Meyerzon, Blumenthal, and Koike

As noted above, the Examiner rejected claims 3, 7-11, 13, 16, 21, 23, 25 under 35 U.S.C. §103(a) as being unpatentable over Meyerzon et al. (U.S. Patent No. 6,638,314), in view further of Blumenthal (U.S. Patent No. 6,026,409), and in further view of Koike et al. (U.S. Patent No. 7,194,678).

Meyerzon discloses a mechanism for obtaining information pertaining to electronic documents

that reside on one or more server computers. In particular, Meyerzon teaches a web crawler program including a gatherer process for gathering information pertaining to electronic documents. See Meyerzon at col. 8, lines 58-60. In the system of Meyerzon, worker threads process URLs and then pass each URL to a filter daemon. See Meyerzon at col. 9, lines 13-16. The filter daemon uses the URL to retrieve the electronic document at the address specified by the URL. See Meyerzon at col. 9, lines 16-20. After retrieving an electronic document, the filter daemon parses the electronic document and returns a list of text and properties. See Meyerzon at col. 9, lines 29-31. The worker thread then passes the list of properties and text to the indexing engine for creating an index which is used by the search engine in subsequent searches. See Meyerzon at col. 10, lines 13-16. A user may then examine the list of documents returned by the search engine, select a document, and then the web browser displays the selected document to the user. See Meyerzon at col. 8, lines 23-25 and 32-35.

On the other hand, amended independent claims 21, 23, and 25 now more clearly recite retrieving, to a server, with a web crawler from a network address, a dynamic data document with client-side scripting code therein;

retrieving, to a server, with a web crawler from a network address, a dynamic data document with client-side scripting code therein;

executing, at the server, a web-browser, as part of the web crawler, wherein the web-browser renders an in-memory copy of the dynamic data document which has been retrieved, wherein the in-memory copy of the dynamic data document maintains a rendered web-browser display format and a rendered web-browser display layout of the dynamic data document when the web-browser renders the in-memory copy of the dynamic data document;

executing, at the server instead of a client system, a browser scripting engine as part of the web-browser, wherein the browser scripting engine executes the client-side scripting code and loads content as directed by the client-side scripting code into the in-memory copy creating a final web-browser display representation of the dynamic data document so that the final web-browser display representation is substantially similar to when the dynamic data document is rendered at a user's web-browser and viewed by a user in the user's web-browser running on the client system when all the dynamic data is viewed; and

indexing, at the server, the content in the memory, wherein the content being indexed is the content which has been loaded by the browser scripting

engine in order to index the dynamic data document as if being viewed by the user in the user's web-browser on the client system.

Support for this amendment can be found in the Specification as originally filed at page 4, lines 28-29 to page 5, lines 1-10; page 10, lines 13-24; and page 12, lines 8-29 to page 13, lines 1-9 and FIGs. 3 – 8. No new matter has been added.

In the present Office Action, the Examiner states on page 2 that Meyerzon teaches:

retrieving, to a server, with a web crawler from a network address, a data document with client-side scripting code therein (Figure 2: Here, a web crawler server is implemented between a client and a web server)

Applicants respectfully point out that the Examiner is ignoring claim elements. For example, the Examiner merely states that Meyerzon shows a web crawler in FIG. 2. The Examiner did not consider nor give any weight the presently recited “retrieving...a data document with client-side scripting code therein”. Meyerzon is silent on a web crawler retrieving a data document with client side scripting code thereon. Accordingly, the presently claimed invention distinguishes over Meyerzon for at least this reason.

The Examiner on page 3 of the Office Action also states that Meyerzon teaches:

Executing, at the server, a web-browser, as part of the web crawler, wherein the web-browser displays an in-memory copy of the data document which has been retrieved, wherein the in-memory copy of the data document maintains a web-browser display format and a web-browser display layout of the dynamic data document when displayed in the web browser (Meyerzon Col 7 Lines 60-65 and Col 8 Lines 15-20: Here, the crawler acts as a web browser in that it requests the web page data. These requested web page documents are stored in memory in a display format)

Applicants respectfully disagree. First of all, Meyerzon merely teaches a web crawler that crawls web pages in a traditional matter. Meyerzon further adds a method for only retrieving web pages that have changed from previous crawls.

Additionally, Applicants have amended the independent claims to more clearly recite:

executing, at the server, a web-browser, as part of the web crawler, wherein the web-browser renders an in-memory copy of the dynamic data document which has been retrieved, wherein the in-memory copy of the dynamic data document maintains a rendered web-browser display format and a rendered web-browser display layout of the dynamic data document when the web-browser renders the in-memory copy of the dynamic data document;

Applicants respectfully request that the Examiner read all remarks and arguments presented herein so that the Examiner can have a clear understanding of the differences between the presently claimed invention and Meyerzon. Meyerzon does not teach a web-browser at the server as part of the web crawler. The Examiner states that “a crawler acts as a web browser in that it requests the web page data”. However, the independent claims state that a web-browser is executed as part of the web-crawler. In other words the web-crawler isn’t merely acting as a web-browser but a separate web browser is being executed as part of the web crawler.

As discussed in the previous Response With Amendment, Meyerzon merely discloses a conventional web crawler that searches remote server computers for electronic documents. The web crawler of the present claimed invention, on the other hand, utilizes a web-browser for displaying an in-memory copy of a dynamic data document. Dynamic data documents comprise executable client side software code that is to be executed by a client’s web-browser. See the background of the present invention. A conventional web crawler cannot properly analyze dynamic data documents. In other words, with a conventional web crawler disclosed by Meyerzon, the summarization process fails or produces flawed results when the document itself contains executable client side software code. The reason for this is that the client side software code (e.g., JavaScript, VBScript, or equivalent) is targeted to be executed and interpreted within a web browser’s scripting engine. Therefore, the presently claimed invention executes, at the server, a web-browser, as part of the web crawler (which is at the server and includes the browser scripting engine). Accordingly, the presently claimed invention distinguishes over Meyerzon for at least these reasons.

Furthermore, the amended independent claims now more clearly recite “wherein the web-browser renders an in-memory copy of the dynamic data document which has been retrieved, wherein the in-memory copy of the dynamic data document maintains a rendered web-browser display format and a rendered web-browser display layout of the dynamic data document when the web-browser renders the in-memory copy of the dynamic data document”. Meyerzon simply does not teach that an in-memory copy of the dynamic data document is rendered. Merely retrieving web pages and parsing the pages to obtain text and formatting, as Meyerzon teaches, does not automatically mean that Meyerzon is teaching these claim elements. As discussed in more detail below and by Meyerzon at col. 9, lines 29-59, Meyerzon is merely parsing HTML code. Therefore, Meyerzon is not rendering an in-memory copy of the dynamic data document which has been retrieved.

The following arguments and remarks include, in part, arguments and remarks that were presented in the previous Response With Amendment and are being repeated because they clearly show how Meyerzon does not teach or suggest “wherein the web-browser renders an in-memory copy of the dynamic data document which has been retrieved, wherein the in-memory copy of the dynamic data document maintains a rendered web-browser display format and a rendered web-browser display layout of the dynamic data document when the web-browser renders the in-memory copy of the dynamic data document”. The presently claimed invention is not merely “construct[ing] a data structure of the webpage in memory” as has been argued by the Examiner in the past. The dynamic data document is copied into the memory of the server and is represented the same way as is if the dynamic data document has been rendered (e.g., creating the finalized webpage to be displayed to the user) and displayed on a screen. The term render is used as known in the art and was not *ipsis verbis* (not in the identical words) in the specification. The Examiner is respectively reminded that this was sufficiently described in the specification, for example, on page 5, lines 1-10, page 12 lines 4-6 and 8-29, and FIG. 6 on page 5 as originally

filed albeit not in the identical words.¹

This ability of the web crawler to create the rendered in-memory copy of the dynamic data document enhances existing document gathering and analysis by, for example, dramatically improving the quality of the extracted metadata. This is due to the fact that the summarization of a document is based on the whole and complete document as it was designed by the document's author; the static heterogeneous data as well as the problematic dynamic data is completely rendered and integrated into the metadata for subsequent indexing of all metadata by a web crawler. For example, a dynamic in-memory representation of the web page, as intended to be seen by an end user, is created to extract the most accurate and comprehensive data set possible. A standard web crawler, as taught by Meyerzon, is not able to compose this type of highly dynamic and distributed document that includes dynamic information such as client side script, applets, or their equivalents.

Meyerzon merely teaches that a web crawler retrieves electronic documents and a filter daemon parses the electronic documents for returning a list of text and properties. Meyerzon explicitly states that text and properties are obtained from tags within the HTML documents. See Meyerzon at column 9, lines 9-43. Therefore, Meyerzon is working on HTML source code, as compared to a “*render[ed] in-memory copy of the dynamic data document which has been retrieved*”. The information is passed to an indexing engine which creates an index of the retrieved data. The index contains reference information and pointers to corresponding electronic documents, for example, keywords. See Meyerzon at col. 8, lines 1-16. Therefore,

¹ If, on the other hand, the specification contains a description of the claimed invention, albeit not in *ipsis verbis* (in the identical words), then the examiner or Board, in order to meet the burden of proof, must provide reasons why one of ordinary skill in the art would not consider the description sufficient. See *In re Alton* (Fed. Cir 1996) (Emphasis Added). See also *Fujikawa v. Wattanasin* (Fed. Cir. 1996), *ipsis verbis*. As the Board recognized, however, *ipsis verbis* disclosure is not necessary to satisfy the written description requirement of section 112. Instead, the disclosure need only reasonably convey to persons skilled in the art that the inventor had possession of the subject matter in question. *In re Edwards*, 568 F.2d 1349, 135152, 196 USPQ 465, 467 (CCPA 1978). See MPEP 2163 subsection II 3 (a), second to last paragraph.

Meyerzon clearly does not teach a “*render[ed] in-memory copy of the dynamic data document which has been retrieved*” as recited for the presently claimed invention. Accordingly, the presently claimed invention distinguishes over Meyerzon for at least these reasons as well.

Even further, Meyerzon does not teach or suggest “...*wherein the in-memory copy of the dynamic document maintains a rendered web-browser display format and a rendered web-browser display layout of the dynamic data document when the web browser renders the in-memory copy of the dynamic data document ...*” As discussed above, Meyerzon is merely working on HTML code, which is the different than a representation of the code itself on a browser “*maintain[ing] a web-browser display format and a web-browser display layout of the dynamic data document when the web-browser renders the in-memory copy of the dynamic data document ...*” Stated differently, Meyerzon is working on the source HTML code, where as the present invention is working on content after the source code is rendered by a web-browser. The two results are often different, especially in the case of dynamic web content. For example, look at the source code of a web page from a dynamic e-commerce site such as www.ebay.com. The static HTML code retrieved from eBay is not the final code used to display the page because of scripting and other dynamic content is loaded only when the web page is displayed in a web browser. Accordingly, the presently claimed invention distinguishes over Meyerzon for at least these reasons as well.

The Examiner further states on page 3 of the Office Action that Meyerzon teaches:

Executing, at the server instead of a client system, a browser scripting engine as part of the web-browser for loading content as directed by the client-side scripting code into the in-memory copy creating a final web-browser display representation of the dynamic data document so that the final web-browser display representation is substantially similar to when the data document is viewed by a user in the user's web-browser running on the client system when all the data is viewed (Meyerzon Col 7 Lines 60-65 and Col 8 Lines 15-20)

Meyerzon at col. 7, lines 60-65 and col. 8, lines 15-20, merely states the following:

The Web crawler program 206 searches remote server computers 218 connected to the network 216 for electronic documents 222 and 224. The Web crawler 206 retrieves electronic documents and associated data. The contents of the electronic documents 222 and 224, along with the associated data, can be used in a variety of ways....The type of information stored in the index depends upon the complexity of the indexing engine, which may analyze the contents of the electronic document and store the results of the analysis.

A client computer 214, such as the personal computer 20 (FIG. 1), is connected to the server computer 204 by a computer network 212....

Meyerzon is only teaching that a web crawler is retrieving electronic documents. Applicants respectfully request the Examiner point out particularly and explain how Meyerzon teaches "Executing, at the server instead of a client system, a browser scripting engine as part of the web-browser for loading content as directed by the client-side scripting code into the in-memory copy creating a final web-browser display representation of the dynamic data document so that the final web-browser display representation is substantially similar to when the data document is viewed by a user in the user's web-browser running on the client system when all the data is viewed".

Applicants have amended the independent claims to make these claims ever clear. The independent claims now more clearly recite:

executing, at the server instead of a client system, a browser scripting engine as part of the web-browser, wherein the browser scripting engine executes the client-side scripting code and loads content as directed by the client-side scripting code into the in-memory copy creating a final web-browser display representation of the dynamic data document so that the final web-browser display representation is substantially similar to when the dynamic data document is rendered at a user's web-browser and viewed by a user in the user's web-browser running on the client system when all the dynamic data is viewed

Meyerzon simply does not teach executing a browser scripting engine as part of the web-browser, or that the "browser scripting engine executes the client-side scripting code and loads

content as directed by the client-side scripting code into the in-memory copy creating a final web-browser display representation of the dynamic data document so that the final web-browser display representation is substantially similar to when the dynamic data document is rendered at a user's web-browser and viewed by a user in the user's web-browser running on the client system when all the dynamic data is viewed". For example, if a webpage requires Java code to be executed at the client in order to display additional data to the user, the browser scripting engine (which is part of the web-browser of the web crawler) at the server executes this Java code in order to include the additional data in the rendered in-memory copy of the webpage. Meyerzon is incapable of performing this claim element because the web crawler of Meyerzon does not include a browser scripting engine or functionalities of a web browser.

Stated differently, the execution of the browser scripting engine allows content to be loaded into memory as directed by the client-side scripting code to create a final web-browser display representation of the dynamic data document. This final web-browser display representation is substantially similar to when the rendered at a user's web-browser and viewed by a user in the user's web-browser running on the client system when all the dynamic data is viewed. Meyerzon merely parses the HTML components of a document to create a list of text and properties. See Meyerzon at col. 9, lines 30-59. Applicants respectfully request the Examiner to point out where in Meyerzon does Meyerzon teach a scripting engine within the web crawler that performs the above claim elements. Accordingly, the presently claimed invention distinguishes over Meyerzon for at least these reasons as well.

The Examiner also states on page 3 of the present Office Action that Meyerzon teaches:

Indexing, at the server, the content in the memory, wherein the content being indexed is the content which has been loaded by the browser scripting engine in order to index the data document as if being viewed by the user in the user's web-browser on the client system (Figures 4-5)

Applicants respectfully disagree. Figures 4 and 5 of Meyerzon **do not** teach or suggest indexing content which has been loaded by the browser scripting engine in order to index the data document as if being viewed by the user in the user's web-browser on the client system. The

Examiner is continuing to ignore elements of the claims such as “...the browser scripting engine executes the client-side scripting code and loads content as directed by the client-side scripting code...” and “...wherein the content being indexed is the content which has been loaded by the browser scripting engine in order to index the data document as if being viewed by the user in the user's web-browser on the client system”. Accordingly, the presently claimed invention distinguishes over Meyerzon for at least these reasons as well.

The Examiner correctly states on page 3 of the present Office Action that “Meyerzon does not specifically mention wherein the server processing unit renders the in-memory webpage prior to analyzing and summarizing the in-memory webpage”. The Examiner goes on to combine Meyerzon with Blumenthal to overcome the deficiencies of Meyerzon. However, the combination of Meyerzon and Blumenthal still does not teach or suggest the presently claimed:

retrieving, to a server, with a web crawler from a network address, a dynamic data document with client-side scripting code therein;

executing, at the server, a web-browser, as part of the web crawler, wherein the web-browser renders an in-memory copy of the dynamic data document which has been retrieved, wherein the in-memory copy of the dynamic data document maintains a rendered web-browser display format and a rendered web-browser display layout of the dynamic data document when the web-browser renders the in-memory copy of the dynamic data document;

executing, at the server instead of a client system, a browser scripting engine as part of the web-browser, wherein the browser scripting engine executes the client-side scripting code and loads content as directed by the client-side scripting code into the in-memory copy creating a final web-browser display representation of the dynamic data document so that the final web-browser display representation is substantially similar to when the dynamic data document is rendered at a user's web-browser and viewed by a user in the user's web-browser running on the client system when all the dynamic data is viewed; and

indexing, at the server, the content in the memory, wherein the content being indexed is the content which has been loaded by the browser scripting engine in order to index the dynamic data document as if being viewed by the user in the user's web-browser on the client system.

Accordingly, the presently claimed invention distinguishes over Meyerzon and Blumenthal alone and/or in combination with one another.

The Examiner correctly states on page 4 of the present Office Action that “Meyerzon does not specifically disclose wherein the data document is a dynamic data document”. However, the Examiner goes on to combine Meyerzon with Koike stating that “Koike discloses a proxy server assembling a dynamic data document for display at a client browser (Figure 6; column 7, lines 13-33). It would have been obvious to one of ordinary skill in the art at the time of the Applicant's invention to have combined Koike with Meyerzon, since it would have allowed a user to more quickly receive the dynamic data”.

Applicants respectfully point out that Koike only discloses “dynamically generating a customized WWW page according to user information and or requests”. See Koike at col. 6, lines 39-41. This is clearly not the same as “a dynamic data document with client-side scripting code therein”. Accordingly, the presently claimed invention distinguishes over Meyerzon and Koike alone and/or in combination with one another.

For the foregoing reasons, independent claims 21, 23, and 25 distinguish over Meyerzon taken alone and/or in view of Blumenthal and/or in view of Koike. Claims 3, 7-11, 13, and 16 depend from claims 21 and 23, respectively, either directly or by way of an intervening claim. Since dependent claims contain all the limitations of the independent claims, claims 3, 7-11, 13, and 16 distinguish over Meyerzon taken alone and/or in view of Blumenthal and/or in view of Koike, as well, and the Examiner's rejection should be withdrawn, which withdrawal is respectfully requested.

Rejection under 35 U.S.C. §103(a) in view of Meyerzon, Blumenthal, Koike, and Hobbs

As noted above, the Examiner rejected claims 4-6 and 17-19 under 35 U.S.C. 103(a) as being unpatentable over Meyerzon et al. (U.S. Patent No. 6,638,314), Blumenthal (U.S. Patent No. 6,026,409), Koike et al. (U.S. Patent No. 7,194,678), and in further view of Hobbs (U.S. Patent

The remarks and arguments above made with respect to independent claims 21, 23, and 25 are also applicable here and will not be repeated.

Hobbs merely teaches a method of dynamically augmenting the contents of file of information on a first network resource. Multimedia information, such as video, audio, graphics and text residing on a plurality of Data Warehouses, RDMS, or ODBA systems is selected. This multimedia information is then linked across the Internet, or other network, to any phrase, work, sentence, numbers, movies, audio, and the like contained in documents on an Internet or intranet web site so that any viewer of a web site can directly access updated information in the Data Warehouse.

The Examiner states on page 7 of the present Office correctly states that “Meyerzon does not specifically (Sic) executing one or more Java applets with textual content embedded thererein”. The Examiner goes on to combine Meyerzon with Hobbs, stating that “Hobbs mentions that Java applets are used (Hobbs Col 28 Line 35). It would have been obvious to one of ordinary skill in the art at the time of the invention, to apply Hobbs to Meyerzon, providing Meyerzon the benefit of using Java Applets for web pages in the process of searching the web documents because Java Applets are compatible with many web pages and browsers.”

Hobbs merely states “A fifth embodiment of the present invention uses a non-CGI programmic object oriented communications protocol employing embedded applications, such as Sun Microsystem's Java applets or Microsoft's Active X combined and coordinated with a scripted language such as Java Script or VBScript. The scripted language would contain, among other things, the authentication arguments; query arguments for all the selected databases connected to the document; the screen positions for all the windows and frames; and controls for the windows and frames.” Just because Hobbs mentions Java does render the presently claimed “wherein the

executing a browser scripting engine as part of the web-browser for loading content as directed by the client-side scripting code into the in-memory copy further comprises executing on the server one or more Java applets with textual content embedded therein” obvious. Applicants respectfully remind the Examiner that the combination of Meyerzon, Blumenthal, Koike, and Hobbs has to teach or suggest that a browser scripting engine is executed by a web crawler for loading content as directed by the client-side scripting code into the rendered in-memory copy and that the browser scripting engine executes one or more Java applets with textual content embedded therein. Meyerzon, Blumenthal, Koike, and Hobbs, either alone or in combination with one another, fail to teach or suggest this claim element. These arguments and remarks are also applicable to dependent claims 5-6 and 17-19, as well. Accordingly, the presently claimed invention and particularly independent claims 21 and 23 and their dependent claims 5-6 and 17-19 distinguish over Meyerzon, Blumenthal, Koike, and Hobbs, either alone or in combination with one another, for at least these reasons and the Examiner’s rejection should be withdrawn, which withdrawal is respectfully requested.

Rejection under 35 U.S.C. §103(a) in view of Meyerzon, Blumenthal, Koike, and Lawrence

As noted above, the Examiner rejected claims 12, 22, and 24 under 35 U.S.C. 103(a) as being unpatentable over Meyerzon et al. (U.S. Patent No. 6,638,314), Blumenthal (U.S. Patent No. 6,026,409), Koike et al. (U.S. Patent No. 7,194,678), and in further view of Lawrence (U.S. Patent No. 6,289,342).

The remarks and arguments above made with respect to independent claims 21, 23, and 25 are also applicable here and will not be repeated.

Lawrence merely teaches an “Autonomous Citation Index autonomously creates a citation index from literature in electronic format [printed literature can be converted to electronic form using optical character recognition (OCR)]. An ACI system autonomously locates new articles, extract

citations, identifies citations to the same article which occur in different formats, and identifies the context of citations in the body of articles. The viability of autonomous citation indexing depends on the ability to perform these functions accurately.” See Lawrence at col. 5, lines 50-58.

Meyerzon, Blumenthal , Koike, and Lawrence alone and/or in combination with one another, do not teach or suggest the presently claimed invention as recited for independent claims 21 and 23 and, therefore, dependent claims 12, 22, and 24. Accordingly, independent claims 21 and 23 and dependent claims 12, 22, and 24 distinguish over Meyerzon, Blumenthal , Koike, and Lawrence alone and/or in combination with one another, and the Examiner’s rejection should be withdrawn, which withdrawal is respectfully requested.

CONCLUSIONS

In light of the Office Action, Applicants believe these amendments serve a useful clarification purpose, and are desirable for clarification purposes, independent of patentability. Accordingly, Applicants respectfully submit that the claim amendments do not limit the range of any permissible equivalents.

Applicants acknowledge the continuing duty of candor and good faith to the disclosure of information known to be material to the examination of this application. In accordance with 37 CFR §§ 1.56, all such information is dutifully made of record. The foreseeable equivalents of any territory surrendered by amendment is limited to the territory taught by the information of record. No other territory afforded by the doctrine of equivalents is knowingly surrendered and everything else is unforeseeable at the time of this amendment by the Applicants and their attorneys.

Applicants respectfully submit that all of the grounds for rejection stated in the Examiner’s

Office Action have been overcome, and that all claims in the application are allowable. No new matter has been added. It is believed that the application is now in condition for allowance, which allowance is respectfully requested.

The Commissioner is hereby authorized to charge any fees that may be required or credit any overpayment to Deposit Account 09-0441. In view of the preceding discussion, it is submitted that the claims are in condition for allowance. Reconsideration and re-examination is requested.

PLEASE, if for any reason the Examiner finds the application other than in condition for allowance, the Examiner is invited to call the undersigned attorney at (561) 989-9811 should the Examiner believe a telephone interview would advance the prosecution of the application.

Respectfully submitted,

Date: February 29, 2008

By: / Jon Gibbons/
Jon Gibbons
Registration No. 37,333
Attorney for Applicants

FLEIT, KAIN, GIBBONS,
GUTMAN, BONGINI & BIANCO P.L.
One Boca Commerce Center, Suite 111
551 Northwest 77th Street
Boca Raton, FL 33487
Tel. (561) 989-9811
Fax (561) 989-9812